drupal.org | Palantir

Scrubbing out SASS

How to Replace Compass

How do we strategically remove Compass from the project without breaking dependency trees or losing valuable functionality in the site?
Let’s start with breaking down *bluecheese/sass/styles.scss*:

```
////////////////////////
// Compass Imports
////////////////////////
@import 'compass';
@import 'compass/reset';
@import 'compass/utilities/general/clearfix';
@import 'compass/typography/text/replacement';
@import 'compass/css3';
@import 'susy';
@import 'breakpoint';

$legacy-support-for-ie6: false;

////////////////////////
// Cheating at CSS
////////////////////////
* {@include box-sizing('border-box');}

// Import styles
@import 'partials/*';
@import 'partials/utilities/**/*';
@import 'partials/common/**/*';
@import 'partials/drupalorg/**/*';
```

`@import 'compass';`

Compass is an open-source CSS authoring framework which uses the Sass stylesheet language to make writing stylesheets powerful and easy.

- Compass on its own (when not being the top of the dependency tree for others) is only used to watch and compile the CSS. We can use Gulp if we determine we will need to maintain any Sass, so Compass will be entirely unnecessary (plus it is deprecated).

```
@import 'compass/reset';
```

These utilities are used to reset your document. The easiest way to use them is to simply @import "compass/reset" which will import this module and apply the appropriate mixins for you.

- I can find zero instances of this in use currently.

```
@import 'compass/utilities/general/clearfix';
```

A clearfix will extend the bottom of the element to enclose any floated elements it contains.

- Currently 32 instances of this exist in 20 separate files, which we can convert to the most modern CSS clear-fix.

```
@import 'compass/typography/text/replacement';
```

Hides text by squishing, sliding, or putting it behind another element.

- I can find zero instances of this in use currently.

```
@import 'compass/css3';
```

The CSS3 module provides cross-browser mixins for CSS properties introduced in CSS3, for example border-radius and text-shadow.

- Although CSS3 was in the experimental stages when this module was developed, and therefore needed special support to create mixins for the SASS logic, we are now in a new era and our CSS has leveled up.

```
@import 'susy';
```

Your markup, your design, your opinions | our math. The only requirement is Sass, but Susy was built to be part of the Compass ecosystem, and we recommend pairing with tools like Breakpoint and Vertical Rhythms.

- Susy will have her own section below.

```
@import 'breakpoint';
```

Breakpoint makes writing media queries in Sass super simple.

- Currently 133 instances of this are in use in 41 separate files, but we can (and should) use media queries as (will be soon) laid out in the CSS layout recommendations.

```
$legacy-support-for-ie6: false;
```

This is not necessary since IE is now deprecated.

```
* {@include box-sizing('border-box');}
```

This syntax is just a workaround to get SASS to perform a task that CSS does natively. We can just include the box-sizing to the root if we need it universally applied, although we should be careful not to apply too many global selectors.

Other Compass References

1. *bluecheese/sass/_glob.scss*
    a. Importing Compass for use in the *styles* doc.
2. *bluecheese/README.txt*
    a. References the CSS which is compiled using Compass with a link to the contribution guide for bluecheese (last updated July 2016).
3. *bluecheese/Gemfile*
    a. Imports the Ruby Gems, which can all be referenced and explained later in this doc.
4. *bluecheese/config.rb*
    a. This doc adds the requirements for any additional Compass plugins, all of which are targeted in Ruby Gems section:
        i. toolkit
        ii. susy
        iii. breakpoint
        iv. sass-globbing
        v. singularitygs
5. The remaining three files which reference Compass are less utilitarian:
    a. *bluecheese/bluecheese.info.yml*: Referred to in the description
    b. *bluecheese/composer.json*: Same description as info file
    c. *bluecheese/Gemfile.lock*: Exact versioning for the Gems

How to Replace Susy

Per Susy's page, the basic quick-start after installing and importing Susy is below:

> *The basic Susy layout is composed using two simple mixins:*
>
> ```
> @include container; // establish a layout context
> @include span(<width>); // lay out your elements
> ```
>
> *For example:*
>
> ```
> body { @include container(80em); }
> nav { @include span(25%); }
> ```
>
> *If you want to lay your elements out on a grid, you can use the span mixin to calculate column widths:*
>
> ```
> nav { @include span(3 of 12); }
> ```

> *But you don’t have to do things the Susy way. We give you direct access to the math, so you can use it any way you like:*
>
> ```
> main {
>   float: left;
>   width: span(4);
>   margin-left: span(2) + gutter();
>   margin-right: gutter();
> }
> ```

Basically Susy is a Ruby Gem (see below) that offers a cohesive layout pattern for the styling. CSS has a wealth of layout options that we can harness that take grids and responsive design into account, so we will just need to narrow down what we choose to use for each part of the site and move forward from there. This will be accomplished, at least partially, in ticket DCM-116.

How to Replace Breakpoint

This, along with Susy, is listed as one of the main SASS plugins on the bluecheese reference page. Getting rid of this dependency is already listed above, however, and is as simple as using CSS media queries.

How to Replace sass-globbing

The Ruby Gem sass-globbing essentially allows SASS users to import an entire directory rather than individual files or even folders. This allows all the styling to bubble up to a top level to be compiled and computed at once.
When we transition to a CSS-only module, globbing won’t be necessary. In fact, it will be counter-productive if we hope to keep our code performant, as splitting the CSS into separate modules and then loading it as needed will keep resource-blocking to a minimum.

How to Replace singularitygs

This is an outdated responsive grid system helper that has little documentation online. This hasn’t been maintained or updated in seven years, and CSS has come a long way in the interim! We can use the CSS grid system for a performant and responsive grid.

How to Replace Ruby Gems

We already did it! Each of the previous headers were “Gems” from Ruby Gems, which is just a package manager (Ruby) for the individual Gems (dependencies). Now that we’ve unpacked the dependencies, we have a path forward to determine how we will use our CSS.